The MagPi magazine is published by Raspberry Pi (Trading) Ltd., Mount Pleasant House, Cambridge, CB3 0RN. ISSN: 2051-9982.

Ohsillyscope

Con l'ingegnoso riff di Thomas McDonald sull'oscilloscopio, le esibizioni di musica dal vivo non solo suonano alla grande, ma Hanno anche un bell'aspetto, come scopre **David Crookes**

MAKER

Thomas McDonald

Thomas si è laureato alla UCCS in Scienza Informatica e Statistica. È un appassionato dell'aria aperta, papà di un cane e drogato di musica.

magpi.cc/ohsillyscope

▲ Thomas voleva un modo per suonare la sua chitarra e vedere una rappresentazione visuale delle melodie sullo schermo

Guarda il video ufficiale del singolo candidato ai Grammy degli Arctic Monkeys, Do I Wanna Know?, del 2013 e vedrai le onde sonore vibrare in sincronia con le percussioni, la chitarra solista e il cantante Alex Turner prima di trasformarsi in un'animazione.

È un effetto semplice ma avvincente e certamente ha colpito gli appassionati di musica, superando più di un miliardo di visualizzazioni su YouTube nel 2020. “La mia parte preferita dell'andare ai concerti sarà sempre la grafica e il video musicale mi ha sicuramente ispirato”, afferma Thomas McDonald. Lo ha portato alla creazione di una versione open source di un oscilloscopio che visualizza l'audio dal vivo come una forma d'onda. Thomas lo chiama "Ohsillyscope".

Il progetto consente a Thomas di suonare strumenti musicali e vedere le melodie visualizzate come onde su un display a matrice di LED 64×64. “Dopo aver preso la matrice LED e averci giocato un po’, ho pensato che ci sarebbe dovuto essere un modo per usarla con la mia chitarra e altri strumenti musicali” spiega. “È sempre molto divertente guardare il segnale dal vivo dal software del sintetizzatore, ma ho immaginato di poter fare qualcosa di più fisico con la matrice”.

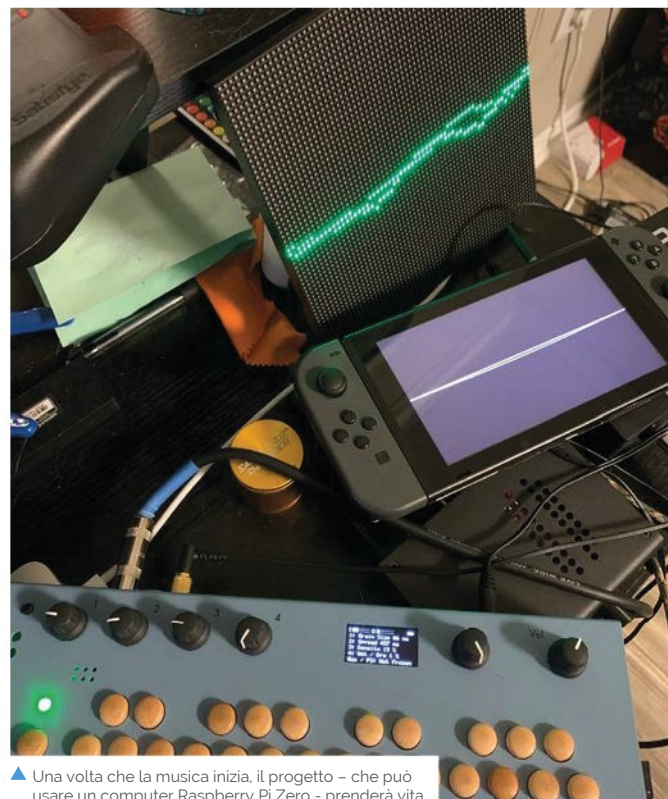
Live jives

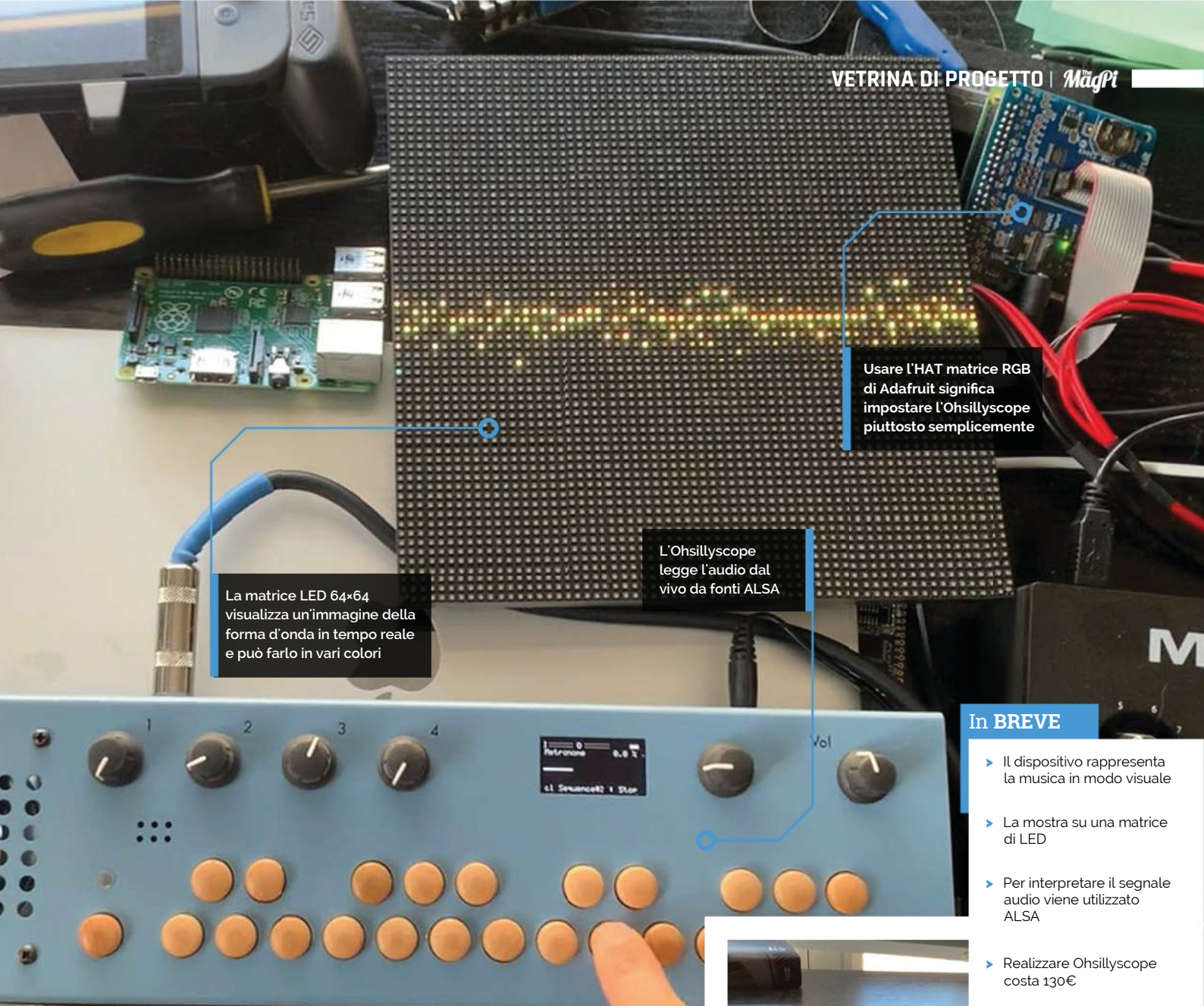
L'idea, spiega, si è ridotta alla registrazione di un segnale audio dal vivo con un computer Raspberry Pi. “In fondo, voglio realizzare immagini complicate che siano reattive al segnale audio in arrivo", dice.

Ha comportato il collegamento del pannello della matrice a un componente aggiuntivo Adafruit RGB Matrix HAT. "Ho Raspberry Pi che riconosce l'interfaccia audio a cui è collegato e ho usato la libreria ALSA (Advanced Linux Sound Architecture) di Linux per la lettura del segnale dall'interfaccia e inserirlo nel codice C che esegue la libreria della matrice LED", afferma Thomas. "Sta campionando alla frequenza di aggiornamento dello schermo, il che restituisce una grafica davvero interessante. Ulteriori informazioni su **alsa-project.org**.

Il risultato è un progetto che può essere preso in considerazione come accompagnamento visivo agli spettacoli musicali. “All'inizio pensavo di poterlo far funzionare tramite MIDI [un protocollo che consente a strumenti musicali, computer e altro hardware di comunicare tra loro], ma l'idea di dover essere collegato al software Ableton Live ha ucciso la motivazione di scrivere una intera libreria per farlo. Volevo qualcosa che non richiedesse di essere davanti a un computer per rappresentare la musica”

Bell’aspetto

Thomas ama usare la sua creazione. “È piccolo e portatile - qualcosa che basta collegare sapendolo

▲ Una volta che la musica inizia, il progetto – che può usare un computer Raspberry Pi Zero - prenderà vita

Usare l'HAT matrice RGB di Adafruit significa impostare l'Ohsillyscope piuttosto semplicemente

La matrice LED 64×64 visualizza un'immagine della forma d'onda in tempo reale e può farlo in vari colori

L'Ohsillyscope legge l'audio dal vivo da fonti ALSA

In BREVE

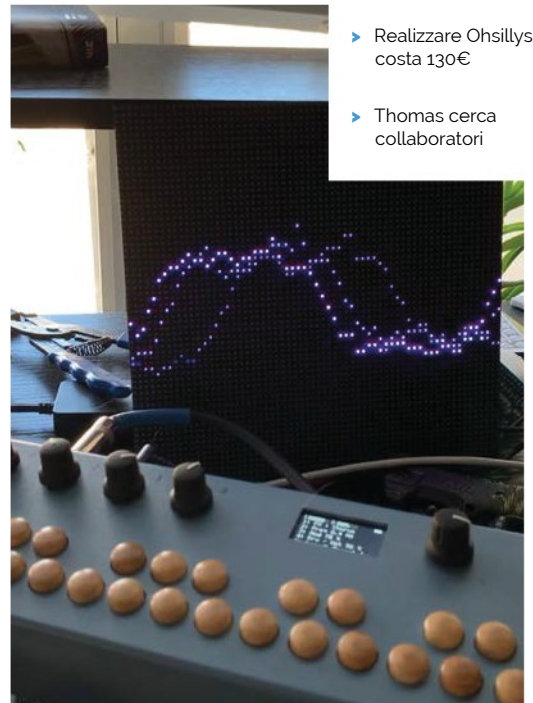
- Il dispositivo rappresenta la musica in modo visuale
- La mostra su una matrice di LED
- Per interpretare il segnale audio viene utilizzato ALSA
- Realizzare Ohsillyscope costa 130€
- Thomas cerca collaboratori

del tutto pronto", dice. "L'HAT a matrice LED è stato creato per Raspberry Pi, quindi è stato un gioco da ragazzi. Mi ha anche aiutato il fatto di avere un paio di computer Raspberry Pi Zero in giro che chiedevano di essere usati."

Attualmente, ha collegato l'Ohsillyscope al sistema PA nella sala prove della sua band. "Raccoglie il basso, chitarre e batteria, creando uno spettacolo interattivo per chiunque guardi" dice.

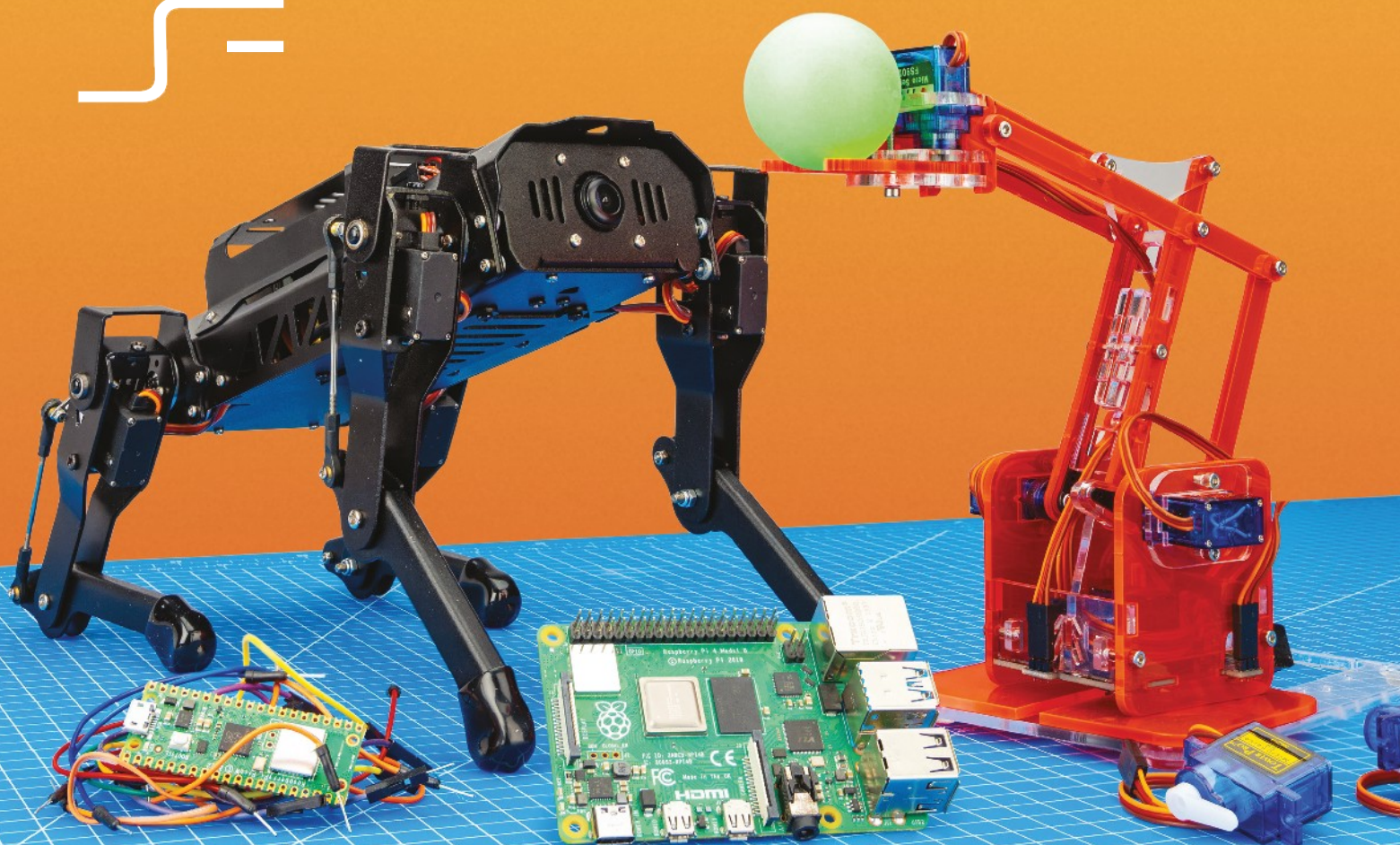
> "Sta campionando alla frequenza di aggiornamento dello schermo"

"Le persone a cui finora abbiamo mostrato l'Ohsillyscope sembrano amarlo e una volta che inizieremo a suonare di più, sicuramente otterremo più attenzione per la band. Siamo decisamente sicuri che starà bene sulla pista da ballo.

Puoi creare la tua versione dell'Ohsillyscope dal momento che il codice è open-source

Python Robots

Creare, controllare e programmare Braccia e gambe robotiche con Python

Di Danny Staple e Lucy Hattersley

TIPI DI SERVO

I robot diventano infinitamente più cool quando usano braccia e gambe. Questa è una caratteristica di tutti, compresi i cani da passeggio, replicanti umanoidi e braccia robotiche prensili.

Tutti questi robot condividono una cosa in comune: i servomotori, noti anche come "servo". Questo tipo di motore ruota avanti e indietro, a differenza dei motori a corrente continua dei robot su ruote che girano all'infinito.

> "Questo speciale ti farà acquisire una comprensione pratica della robotica articolata"

Immergiti in un viaggio ricco di programmazione Python e robotica con questo speciale. Ti spiegheremo come funzionano i servomotori, come puoi controllarli con Raspberry Pi e dare un occhio alcuni dei fantastici kit robotici in commercio.

Che tu sia un hobbista o un aspirante esperto di robotica, questa guida ti fornirà una comprensione pratica del mondo della robotica articolata. Quindi è il momento di attrezzarsi, assembla i tuoi strumenti e intraprendi una emozionante avventura robotica!

I servomotori sono un tipo di motore che ruota avanti e indietro con precisione. Sono usati nei robot che richiedono un controllo preciso, come ad esempio braccia robotiche e cani con le zampe e robot che camminano. Le opzioni più costose includono un sensore di qualche tipo per misurare la posizione. Ecco alcuni tipi di servo a cui prestare attenzione...

SG (Servo Gear)

I servoriduttori come l'SG90 e il più recente SG92 sono economici (circa 4€) e leggeri con un controllo preciso del movimento. L'SG92R può ruotare di circa 180 gradi (90 in ciascuna direzione). Li troverai in progetti entry-level di robotica dove costo e peso sono una criticità.

magpi.cc/sg92

MG (Metal Gear)

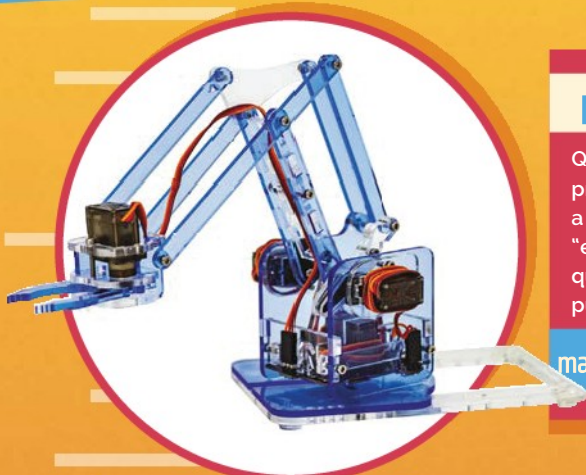
Il prossimo passo avanti nei servi è quello di cambiare gli ingranaggi di plastica in metallo. Servi come questo MG92 (circa 7€) offrono maggiore durata e resistenza rispetto ai modelli SG entry-level, e sono migliori dove il servo dovrà sollevare pesi o sposterà carichi. Sono un po' più pesanti e rumorosi delle loro controparti in plastica.

magpi.cc/mg90

Brushless

In un motore tipico, le spazzole vengono utilizzate per cambiare la polarità del campo magnetico, cosa necessaria affinché il motore continui a girare. Tuttavia, le spazzole sono soggette a usura e possono creare rumore elettrico, limitando la durata e le prestazioni del motore. I motori senza spazzole durano più a lungo, usano meno energia, creano meno calore e fanno meno rumore pur essendo più veloci e con una portata maggiore. Costano molto di più, però.

magpi.cc/brushlessservo

KIT ROBOTICI

MeArm Maker Kit

Questo è il kit che stiamo usando per spiegare il servocomando per la parte principale di questo speciale. MeArm Maker (50$ / 45€) è un braccio robotico a basso costo che può essere assemblato con una semplice chiave esagonale "e un pizzico di entusiasmo". Il kit include parti in acrilico tagliate al laser e quattro servomotori. È semplice, ma quella semplicità lo rende un ottimo punto di partenza prima di passare a robot ambulanti più complessi.

magpi.cc/mearmmaker

Marty v2

Marty the Robot V2 (449$ / 410€) è un affascinante robot a due gambe che dà vita alla programmazione. Sviluppato da Robotical, presenta molti progetti educativi di accompagnamento e questo simpatico robot è controllato tramite Scratch o Python. Robotical afferma che è adatto a bambini dai sette anni in su e fino al livello universitario. Marty è controllato usando carte colorate, un'app per Android o iPhone o Raspberry Pi. Nonostante il suo aspetto carino, Marty è ricco di potenzialità, il che lo rende un ottimo strumento per l'apprendimento e il divertimento.

magpi.cc/marty

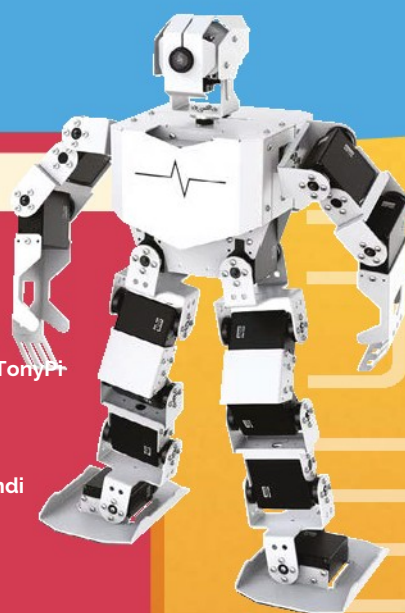
Petoi Bittle X

Abbiamo testato Petoi Bittle (256$ / 233€) e l'abbiamo trovato carino, cane robot delle dimensioni di un palmo che è un ottimo strumento per promuovere l'apprendimento STEM. Un nuovo modello Bittle X è in fase di test e ora offre interazione vocale. È un robot con quattro zampe controllabili e il suo prezzo è un'alternativa più conveniente di animali robot di fascia alta. Viene fornito con la scheda controller NyBoard 1 e dieci servomotori. È un modo delizioso di possedere un cane robot e ci sono molti progetti di programmazione da trovare qui.

magpi.cc/petoibittle

Tony Pi

Hiwonder ci ha impressionato con la sua vasta gamma di robot articolati di alta qualità. TonyPi (570$ / 520€) è un kit umanoide alimentato da Raspberry Pi 4B 4GB. Non è economico, ma la costruzione è superba ed è dotato di servi bus ad alta tensione. Questo avanzato robot è progettato per fornire un'esperienza pratica nell'intelligenza artificiale e nella robotica. Le capacità di TonyPi includono il riconoscimento e il tracciamento delle immagini, rendendolo uno strumento versatile per l'apprendimento e l'esplorazione. Hiwonder fornisce un'interfaccia per app e puoi immergerti nel kit tramite SSH e programmarlo direttamente, quindi va bene sia per i principianti che per gli esperti.

magpi.cc/tonypi

SpiderPi

L'Hiwonder SpiderPi (600$ / 550€) è un robot notevole, in piedi è quasi 30 cm da terra e si allunga a quasi 60 cm quando le sue gambe sono completamente estese. Costruito con 18 servi simili al suo fratello Hiwonder, il TonyPi, lo SpiderPi è controllato da un Raspberry Pi 4 da 4 GB. Questo robot hexapod è dotato di una videocamera HD che può ruotare di 140 gradi, consentendogli di rilevare i volti e segnala il riconoscimento agitando una zampa. SpiderPi esegue vari movimenti tra cui correre in avanti, indietro e lateralmente, oltre a torcere e sollevare le zampe anteriori. La sua fotocamera e la velocità di movimento possono essere regolate tramite una barra scorrevole nell'app. Il robot può anche seguire una linea ondulata sul pavimento, distinguere tra oggetti di diverso colore, eseguire comandi emessi tramite uno script Python legato a un Qrcode. Ha spaventato a morte il nostro gatto.

magpi.cc/spiderpi

Mini Pupper

Mini Pupper di MangDang (da 449$ / 410€) è un kit cane robot ROS open-source progettato per rendere la robotica più accessibile. MangDang afferma che Mini Pupper è un eccellente strumento pratico per l'istruzione STEM e K12. Con questo kit, l'apprendimento della robotica diventa un viaggio divertente e coinvolgente.

magpi.cc/minipupper

ROBOTICA SUPER SEMPLICE

Abbiamo una guida per i principianti della robotica sul numero 120 di *The Magpi*. A differenza delle articolate meraviglie presenti in questo tutorial, il numero precedente mostra come creare un robot su ruote utilizzando parti a basso costo e controllato con codice. I robot su ruote sono molto più semplici, ma meno divertenti.

bit.ly/MagPi120It

Controllare un braccio robotico con Python

Costruisci e controlla un braccio robotico, mentre impari come comandare a distanza i servomotori su Raspberry Pi Pico W

Di Danny Staple

MAKER

DANNY STAPLE

Ha un laboratorio domestico di robot. Li costruisce con Raspberry Pi e Pico come Orionrobot. Autore di *Robotics at Home with Raspberry Pi Pico*.

orionrobots.co.uk

COSA SERVE

- MeArm Maker **magpi.cc/mearmmaker**
- Breadboard **magpi.cc/breadboardhalf**
- Cavetti jumper **magpi.cc/jumperwires**
- Laptop, tablet, o smartphone con Wi-Fi
- Un piccolo cacciavite con testa a croce
- Thonny

Gradi di libertà al gomito, spalla e base

Una pinza utilizza un quarto servomotore

I servomotori consentono al tuo Raspberry Pi Pico di controllare robot articolati, muovendo i loro giunti in posizioni specifiche.

In questo tutorial imparerai dove è possibile utilizzare i servomotori e come collegare un braccio robotico servocomandato a un Raspberry Pi Pico W.

Imparerai come utilizzare la libreria asyncio (**magpi.cc/asyncio**) per controllare i servomotori assieme, rendendo possibile la robotica articolata.

Potrai quindi controllare questo braccio robotico da una rete pagina web o da uno smartphone.

01 Cosa sono i servo motori?

I robot articolati utilizzano servomotori per controllare le loro articolazioni. I servomotori combinano un motore, un riduttore e circuiti di controllo. Di solito son limitati a 180 gradi di movimento.

Un servomotore utilizza il controllo a circuito chiuso e è controllato con modulazione di larghezza di impulso o 'PWM', dove la lunghezza degli impulsi contiene le informazioni sulla posizione.

Un Raspberry Pi Pico può emettere questi segnali, con 65.536 possibili valori. I servomotori rispondono solo a un range di 8000 valori, con impulsi molto brevi 50 volte al secondo. Il servo trova la differenza tra la sua posizione di ingresso e quella di uscita, quindi sposta il motore per ridurre la differenza.

02 A proposito di MeArm

Il MeArm è un kit per braccio robotico economico, tagliato al laser di Mime Industries (**mime.co.uk**). MeArm è progettato da far costruire e programmare a hobbisti e studenti. Mime Industries ha modelli MeArm con elettronica progettata per Raspberry Pi, Arduino e micro:bit. Tuttavia, il MeArm Maker è un modello che consente la connessione a qualsiasi controller e funzionerà bene con Raspberry Pi Pico.

Robot alternativi

È possibile utilizzare le tecniche di questo tutorial per altri robot articolati come Petoi Bittle X, l'Hiwonder TonyPi, o per costruire un esapode o un tentacolo robotico!!

Gradi di libertà

Quando si parla di robot segmentati, come questo braccio, usiamo il termine "gradi di libertà", o DOF, per contare il numero di posti o segmenti di assi che possono muoversi.

Il MeArm utilizza quattro servomotori che gli consentono di muoversi su tre assi e raccogliere le cose. I tre gradi di libertà sono la base rotante, l'articolazione della spalla e l'articolazione del gomito. Gli assi sono comandati da servomotori.

03 Preparare Pico W

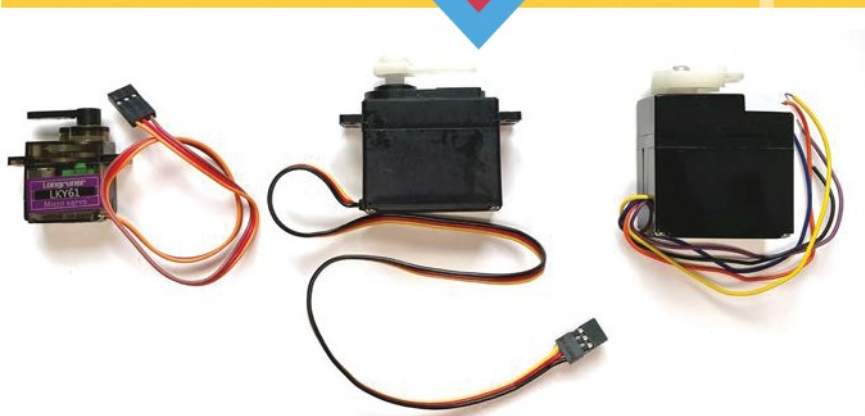
Prima di poter interagire con il braccio, devi preparare Raspberry Pi Pico W. Prima dovrai saldare il connettore con i pin, a meno

> “MeArm è un kit per braccio robotico economico, tagliato al laser”

che tu non abbia comprato un modello con connettore pre-saldato Vedi **magpi.cc/soldering** per aiutarti con la saldatura. Avrai anche bisogno della versione corrente di MicroPython per Raspberry Pi Pico W. Scarica l'ultima immagine UF2 da **magpi.cc/micropython**. Tieni premuto il pulsante BOOTSEL mentre colleghi il Pico W a un computer tramite USB; puoi quindi trascina e rilascia l'immagine UF2 su Pico W sul desktop.

04 Cablare i servo sul Pico W

I servomotori dovranno essere configurati prima di assemblare il braccio. Scollega il Raspberry Pi Pico W dalla porta USB. Ogni servomotore si collega a uno dei connettori a 3 pin sulla scheda MeArm, con il filo arancione o giallo collegato al pin contrassegnato con O o Y sulla scheda.

Il connettore a 6 pin è cablato al Raspberry Pi Pico W. Collega il Pico W a una breadboard e quindi esegui un ponticello da ciascuna connessione etichettata sul connettore a 6 poli alla riga corrispondente della breadboard per l'etichetta sul Pico W.

Una selezione di servomotori. Ci sono di diverse dimensioni per gestire vari compiti. Il robot MeArm usa quelli minuscoli

05 Controllare i servo motori con MicroPython

Collega il Pico W, quindi apri Thonny. Collega Thonny al Pico W e avvia la shell di MicroPython.

Digita le seguenti righe:

```
import machine
grip = machine.PWM(machine.Pin(4, machine.
Pin.OUT))
grip.freq = 50
grip.duty_u16(5000)
```

5000 corrisponde alla posizione centrale per il motore. Puoi provare altri valori e dovresti vedere la rotazione del motore. Tuttavia, assicurati che sia nella posizione 5000 prima di terminare questo passaggio.

Ripeti questa operazione per gli altri tre servo sui pin 5, 6 e 7. Evita di far girare i servomotori a mano, poiché questo potrebbe rompere gli ingranaggi al loro interno.

06 Assemblare MeArm Maker

Con il Pico ancora alimentato, segui le istruzioni MeArm linkate di seguito per collegare le squadrette dei servo mentre i servomotori sono in posizione mediana. Puoi giocare con le posizioni dei servomotori qui, ma assicurati di riportarle a posizione 5000 prima del montaggio.

Quindi, scollega la connessione USB dal Pico, quindi scollega i servomotori dalla scheda del MeArm. Lascia connessi la scheda MeArm e il Pico W. La costruzione del braccio è spiegata al meglio nella documentazione MeArm, che puoi trovare su **magpi.cc/buildingmearm**.

07 Controllare i movimenti del braccio

Il codice per controllare in PWM i servomotori è nella classe Servo in **mearm.py**. È utile vedere il movimento del servo come angoli. La posizione centrale è 0, quindi l'intervallo è +/- 90.

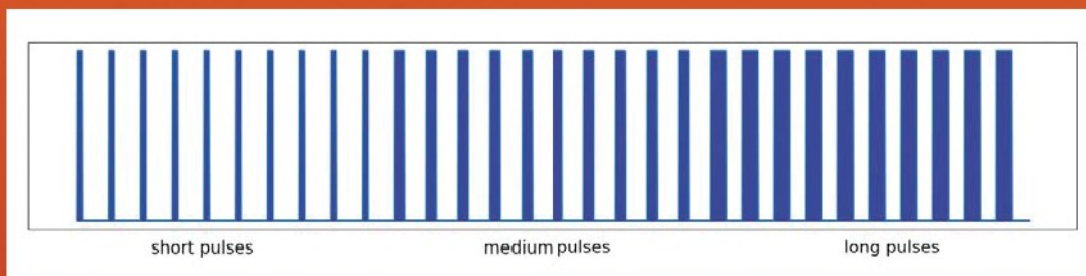
Il valore medio del ciclo di lavoro è 5000 per Pico MicroPython: lo chiameremo `PWM_MID`. Se prendiamo +/- 4000 come intervallo da questa posizione, noi possiamo dividere 4000 per 90 per ottenere un fattore di scala `DEGREES_TO_PWM`.

Possiamo quindi moltiplicare un angolo per questo fattore di scala e aggiungere il risultato a `PWM_MID` per ottenere il ciclo di lavoro.

08 Fare un braccio dai servi

La classe Arm utilizza la classe Servo per ciascuna giunzione. La classe Arm ha numeri di pin predefiniti corrispondenti al cablaggio, ma con numeri di pin diversi, consente due o più braccia! Carica mearm.py in Thonny e premi il pulsante play. Tornerai alla shell MicroPython. Digita `arm.base.set_angle(-30)`, per muovere il braccio base a -30 gradi. Puoi provare questo con ogni giunto. Trova i limiti aumentando i numeri dalla posizione 0 in passi di 5 gradi fino a quando l'articolazione non si muove più, quindi torna indietro di un passo. Non lasciare un giunto oltre questi limiti per evitare danni.

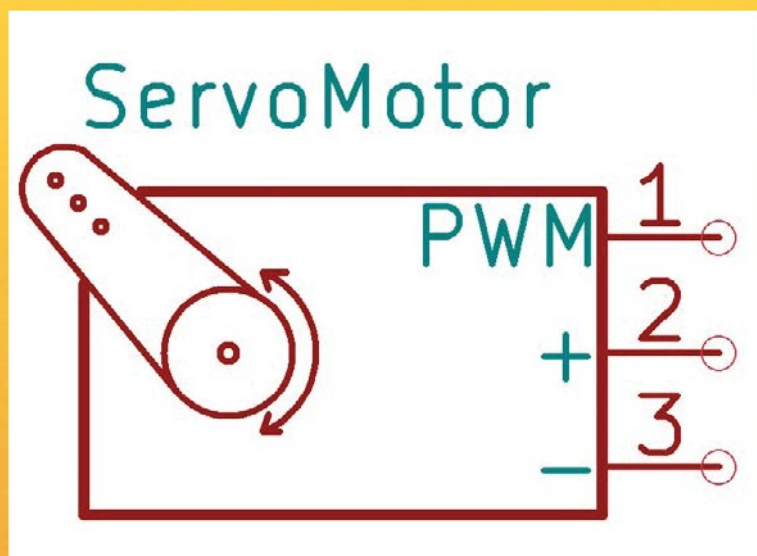
09 Movimento più fluido

I servocomandi saltano, ma possiamo farli muovere in maniera lineare. In `Servo.move`, controlliamo la velocità

Questo grafico visualizza il PWM (modulaszione della ampiezza dell'impulso). Le parti ombreggiate mostrano un segnale alto. Gli impulsi più brevi girano il servo in senso antiorario, quelli medi lo portano al centro e gli impulsi lunghi lo fanno girare in senso orario

e la scorrevolezza con un tempo in secondi e con il conteggio dei passi.

La classe Servo memorizza una posizione `current`; `Servo.move` fissa un obiettivo e fa la differenza tra questo e la posizione `current`. `Servo.move` divide il tempo in passi per `step_time` e divide la differenza di movimento dei passi per `step_size`.

Esegue un ciclo sui passi, aggiungendo `step_size` alla posizione `current`, spostando il servo nella nuova posizione e si ferma poi per il tempo `step_time`.

In Shell, digita `do(arm.elbow.move(30))` per vedere il gomito spostarsi dolcemente a 30 gradi.

10 Muovere le articolazioni assieme

Muovere le articolazioni insieme è più interessante. Abbiamo usato un `do` helper intorno a `Servo.move` perché `move` è asincrono. Quando incontra l'istruzione `await`, può essere eseguito altro codice, abilitandoci a svolgere più attività contemporaneamente. Il metodo `move_together` in `Arm` imposta un processo di movimento per

> "È utile vedere i movimenti del servo come angoli"

ciascuna posizione articolare specificata, e poi usa `await uasyncio.sleep(seconds)` per attendere il completamento di tutte le attività di spostamento. Tutti i task hanno il tempo di essere eseguiti, così le varie possono muoversi insieme.

Prova `do(arm.move_together(base=40, shoulder=20, grip=30, base=50))` per vedere come tutto si muove allo stesso tempo! Forse possiamo fare qualcos'altro?

11 Connessione alla rete

Possiamo connettere il nostro Pico W a una rete Wi-Fi. Aggiungi un file secrets.py su Pico con l'SSID e la password Wi-Fi.

```
SSID = "<network name>"
PSK = "<network password>"
```

All'inizio del file **web_arm.py** vengono importati e usati per connettersi al Wi-Fi. Useremo anche il server web microdot per MicroPython. Puoi installarlo da Thonny con Strumenti > Gestisci pacchetti. Cerca microdot, quindi fai clic su Installa.

Quando esegui **web_arm.py**, cercherà una connessione. Una volta connesso, visualizzerà l'indirizzo di rete del tuo Pico W sulla console.

Un servo di solito ha tre pin: '+' per tensione, '-' per la terra, e giallo o arancione per il segnale di ingresso PWM

12 Creare endpoint di controllo

Il web server microdot ci permette di costruire piccoli endpoint per controllare il nostro braccio. Esiste un oggetto app microdot, che ci consente di impostare un collegamento tra un percorso di richiesta in entrata e una funzione Python, ad esempio collegando `/set_grip/<position>` alla funzione `set_grip`. La parte `<position>` specifica un parametro di posizione che la funzione può utilizzare e passare al braccio.

Anche queste funzioni sono asincrone, quindi il server può gestire le richieste mentre il braccio è ancora in movimento. Carica **secrets.py**, **mearm.py** e **web_arm.py** nel Pico W, quindi fai clic sul pulsante Esegui su **web_arm.py** per avviare il server.

13 Provare gli endpoint

Punta un browser web all'indirizzo mostrato nel passaggio 11 e un movimento da eseguire. Ad esempio: **http://192.168.0.136/set_elbow/30** per spostare il gomito a 30 gradi.

Non devi aspettare che termini un precedente movimento prima di inviare altre istruzioni di movimento al robot. Non si comporterà correttamente se si cerca di muovere lo stesso motore due volte prima che sia finito il primo movimento, però. Questo va bene, ma non è il modo più divertente con cui interagire con il robot.

Le tecniche di questo articolo possono essere utilizzate per controllare i dieci servomotori del cane robot Petoi Bittle X

C'è un altro endpoint in web_arm.py, denominato `/` e connesso a una funzione `index` che serve una pagina per controllare il braccio.

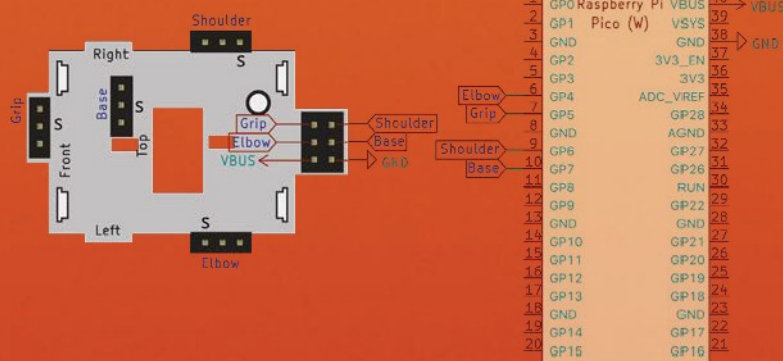
14 Mettere i cursori a una pagina web

La funzione index carica un file control.html e lo rimanda indietro. Il corpo di control.html ha un titolo e quattro elementi `input type="range"`. Ciascun intervallo di input consente di specificare i valori minimo, massimo e iniziale.

L'elemento datalist `snap_to_zero` viene usato per elencare le proprietà degli intervalli per disegnare i marcatori a 0.

La pagina ha una sezione di stile, `overflow: hidden`, che impedisce lo scorrimento accidentale della pagina su un telefono, Le impostazioni di larghezza e altezza fanno in modo che i cursori si adattino bene a qualsiasi schermo, insieme allla impostazione del viewport nell'intestazione della pagina.

Se vai all'indirizzo del tuo Pico W, dovresti vedere i quattro cursori.

Cablaggio del MeArm al Raspberry Pi Pico W. I fili collegano le etichette con lo stesso nome

15 Collegare i cursori

In **control.html**, carica `jquery`, una libreria JavaScript progettata per aiutare con la gestione degli eventi della pagina e l'invio di dati.

Quando il documento è pronto, il nostro JavaScript cerca gli elementi di input e li collega ai gestori di cambiamento per ciascuno di essi.

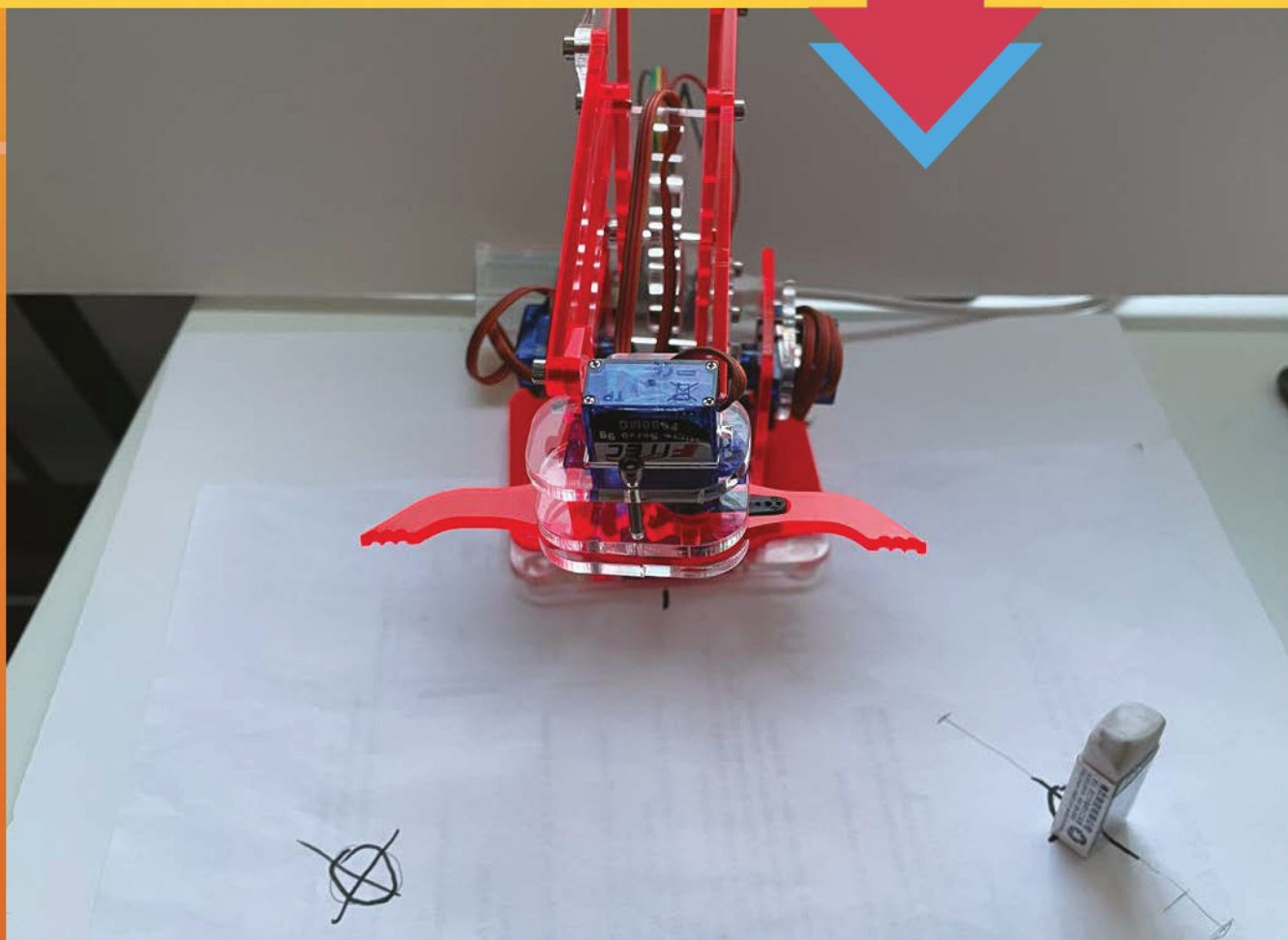
Prendendo ad esempio l'elemento della spalla, quando questo input viene modificato, la funzione invierà una richiesta GET al server **web_arm.py** server. Questo passerà a **/set_base/** e lo combinerà con il valore dell'elemento spalla. L'evento di modifica si verifica quando un input sulla pagina viene spostato e il dito o il pulsante del mouse viene sollevato. Ciò significa che puoi trascinare i cursori per spostare le posizioni del braccio. Puoi anche connettiti con uno smartphone per pilotarlo!

> "Puoi trascinare i cursori per cambiare posizione al braccio"

16 Registrare i passi

Nella parte inferiore della pagina sono presenti i pulsanti per aggiungere passaggi di movimento. Nell'HTML, hanno gestori onclick da chiamare in **web_arm.py**.

Quando si fa clic sul pulsante "add step", **web_arm.py** raccoglie la posizione corrente di ciascun servomotore e memorizza le loro posizioni nella variabile `step_list`. Poi, quando si clicca "run steps", un'attività asincrona chiama `move_together` con ogni gruppo di posizioni, riproducendo l'intera lista.

▲ Tracciare i segni per gli oggetti da raccogliere e spostare con il braccio

Ci sono anche pulsanti di salvataggio e caricamento. Questi memorizzeranno i passaggi in un file JSON sul Pico W, così puoi poi ricaricarli più tardi.

17 Divertiti con i passi registrati

Posiziona il robot sul lato lungo di un foglio di carta. Disegna una linea davanti alla base per allineare la base del braccio. Disegna un segno tra 30 e 60 gradi a sinistra e da 100 a 140 mm dal base del braccio, e un altro segno a destra.

Posiziona un oggetto leggero, come una piccola gomma, su un segno. Utilizzando l'interfaccia web, aprire le pinze, ruota il braccio per afferrare l'oggetto, quindi sposta il braccio in modo che l'oggetto si trovi tra le pinze aperte. Premi il pulsante "add step". Chiudi le pinze e premi "add step".

Ora imposta le articolazioni del gomito, della base e della spalla su 0 e premi "add step". Quindi muovi la base, il gomito, la spalla per mettere l'oggetto sull'altro segno e "add step". Apri le pinze, quindi "add step". Ora sposta il gomito e la spalla verso l'alto ancora "add step". Infine, imposta i giuntidel gomito, base e la spalla a 0 e premi "add step".
Se rimetti l'oggetto sul primo segno e premi il pulsante "run steps", il robot ora raccoglierà automaticamente l'oggetto e lo sposterà sull'altro segno. Puoi usare "save steps" e "load steps" per salvarlo per dimostrazioni!

18 Cosa fare dopo?

Alimentare il servomotore della pinza attraverso un sensore di corrente, come un INA219, così da poter rilevare quando le pinze stringono qualcosa, serrandole fino a quando la corrente non va sopra una soglia.

I robot articolati utilizzano la cinematica inversa per calcolare gli angoli del servo in base alla posizione desiderata.
L'aggiunta di questo renderebbe più facile posizionare il braccio con i controlli.

Potresti aggiungere un sensore di distanza nella parte anteriore del braccio, sposta la base per cercare l'oggetto e combinarlo con la cinematica inversa per pinzare un oggetto posizionato arbitrariamente.

mearm.py

> Linguaggio: **MicroPython**

SCARICA IL CODICE COMPLETO:

```
001.  """
002.  import sys
003.  del sys.modules["mearm"]
004.  from mearm import arm, do
005.  # tests
006.
007.  do(arm.move_together(base=30, shoulder=40,
      elbow=30, grip=20))
008.  do(arm.move_together(shoulder=0, elbow=-30))
009.  """
010.  import machine
011.  import uasyncio
012.
013.
014.  PWM_MID = 5000
015.  PWM_FREQ = 50
016.  DEGREES_TO_PWM = 4000 / 90
017.
018.
019.  class Servo:
020.      def __init__(self, pin):
021.          self.pwm = machine.PWM(machine.Pin(pin,
      machine.Pin.OUT))
022.          self.pwm.freq(PWM_FREQ)
023.          self.current = 0
024.
025.      def set_angle(self, angle):
026.          self.pwm.duty_u16(int(PWM_MID + (
      angle * DEGREES_TO_PWM)))
027.
028.      async def move(self, position, seconds=1,
      steps=100):
029.          step_time = seconds/steps
030.          step_size = (
      position - self.current) / steps
031.          for n in range(steps):
032.              await uasyncio.sleep(step_time)
033.              self.set_angle(self.current)
034.              self.current += step_size
035.
036.
037.  class Arm:
038.      def __init__(self, elbow_pin=4, grip_pin=5,
      shoulder_pin=6, base_pin=7):
039.          self.grip = Servo(grip_pin)
040.          self.elbow = Servo(elbow_pin)
041.          self.shoulder = Servo(shoulder_pin)
042.          self.base = Servo(base_pin)
043.
044.      async def move_together(self, base=None,
      shoulder=None, elbow=None, grip=None, seconds=1,
      steps=100):
045.          if base is not None:
046.              uasyncio.create_task(self.base.move(
      base, seconds=seconds, steps=steps))
047.          if shoulder is not None:
048.              uasyncio.create_task(self.shoulder.
      move(shoulder, seconds=seconds, steps=steps))
049.          if elbow is not None:
050.              uasyncio.create_task(self.elbow.
      move(elbow, seconds=seconds, steps=steps))
051.          if grip is not None:
052.              uasyncio.create_task(self.grip.
      move(grip, seconds=seconds, steps=steps))
053.          await uasyncio.sleep(seconds)
054.
055.
056.  arm = Arm()
057.  do = uasyncio.run
```

web_arm.py

> Linguaggio: **MicroPython**

```
001.  import machine
002.  import uasyncio
003.  import json
004.
005.  from microdot_asyncio import Microdot
006.  from mearm import arm
007.
008.  import network
009.  from utime import sleep
010.
011.  from secrets import *
012.
013.  def wifi_connect():
014.      wlan = network.WLAN(network.STA_IF)
015.      wlan.active(True)
016.      wlan.connect(SSID, PSK)
017.      while wlan.isconnected() == False:
018.          print('In attesa di connessione...')
019.          sleep(1)
020.      print(wlan.ifconfig())
021.
022.  app = Microdot()
023.  step_list = []
024.
025.  @app.route('/')
026.  async def index(request):
027.      with open("control.html") as content:
028.          return content.read(), 200, {
      'Content-Type': 'text/html'}
029.
030.  @app.route('/set_grip/<position>')
031.  async def set_grip(request, position):
032.      uasyncio.create_task(
      arm.grip.move(int(position)))
033.      return 'Moving'
034.
035.  @app.route('/set_base/<position>')
036.  async def set_base(request, position):
037.      uasyncio.create_task(arm.base.move(
      int(position)))
038.      return 'Moving'
039.
040.  @app.route('/set_shoulder/<position>')
041.  async def set_shoulder(request, position):
042.      uasyncio.create_task(arm.shoulder.move(
      int(position)))
043.      return 'Moving'
044.
045.  @app.route('/set_elbow/<position>')
046.  async def set_elbow(request, position):
          uasyncio.create_task(arm.elbow.
047.  move(int(position)))
048.      return 'Moving'
049.
050.  @app.route('/add_step')
051.  async def add_step(request):
052.      state = (arm.base.current, arm.shoulder.
current, arm.elbow.current, arm.grip.current)
053.      step_list.append(state)
054.      return 'Added step'
055.
056.  async def run_steps():
057.      try:
058.          for step in step_list:
059.              await arm.move_together(
*step, seconds=1)
060.      except:
061.          print("Steps were:", step_list)
062.          raise
063.
064.  @app.route('/run_steps')
065.  async def handle_run_steps(request):
066.      uasyncio.create_task(run_steps())
067.      return 'Running steps'
068.
069.  @app.route('/save_steps')
070.  async def handle_save_steps(request):
071.      with open("steps.json", "w") as f:
072.          json.dump(step_list, f)
073.      return 'Saved steps'
074.
075.  @app.route('/clear_steps')
076.  async def handle_clear_steps(request):
077.      step_list.clear()
078.      return 'Cleared steps'
079.
080.  @app.route('/load_steps')
081.  async def handle_load_steps(request):
082.      global step_list
083.      with open("steps.json") as f:
084.          step_list = json.load(f)
085.      return 'Loaded steps'
086.
087.  try:
088.      wifi_connect()
089.      print("Avvio dell'app")
090.      app.run(port=80)
091.  except KeyboardInterrupt:
092.      machine.reset()
```

Trappole Fotografiche Raspberry Pi

Fai foto – e video! - delle tue bestiole locali

Di **Rob Zwetsloot**

MONOFILM A600 300 MONOFILM A600

I moduli fotocamera Raspberry Pi sono meravigliosi, e li abbiamo usati per innumerevoli progetti sulle pagine di *The MagPi* nel corso degli anni. Li programmi direttamente, quindi sono incredibilmente personalizzabili, potendo anche comunicare con il web o la moltitudine di sensori disponibili per Raspberry Pi.

Con tutto questo potere e un po' di... impermeabilità, è incredibilmente facile usare un Raspberry Pi per dare un'occhiata fuori e scoprire il mondo naturale che ti circonda. Qui ci sono alcuni progetti per iniziare...

Galleria di Ricci

Le migliori foto della migliore fauna selvatica scattate dal miglior microcomputer

MONOFILM A600 300 MONOFILM A600

Rob ama sfacciatamente i ricci grazie un simpatico esemplare blu e poiché è una sua caratteristica, ha deciso di avere una pagina dedicata a questi antichi, spinosi e carini abitanti del giardino .

Foto di The Jolly Geographer e Wildlife Trust BCN

Foto di Keith Freeman (@BiscuitKeith)

Foto di @adeibiza

▲ "Ecco muso e dorso di Nora il riccio, catturati da una fotocamera My Naturewatch Raspberry Pi"

◀ "I ricci alla mangiatoia per gatti; ho stampato in 3D alcuni gradini che a volte utilizzano"

▶ "L'ultimo video musicale di Hog Alley. Altri possono essere trovati @hogcamb (principalmente su Instagram). Non è certo che mi abbiano mantenuto sano di mente durante il lock-down, ma certamente occupato.

Progetti Fototrappola

Costruisci la tua fotocamera naturalistica economica

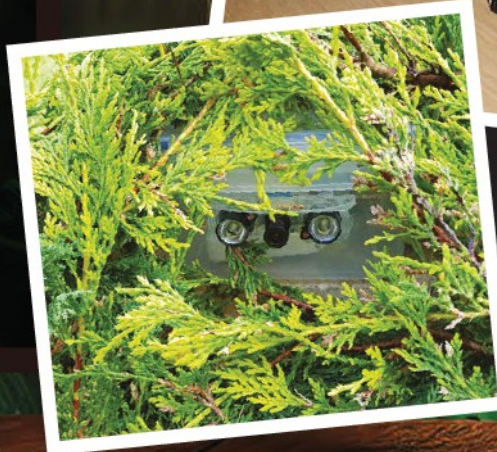
Di **Rob Zwetsloot**

MONOFILM A600 300 MONOFILM A600

Simple Raspberry Pi Camera Trap

magpi.cc/simplecamtrap

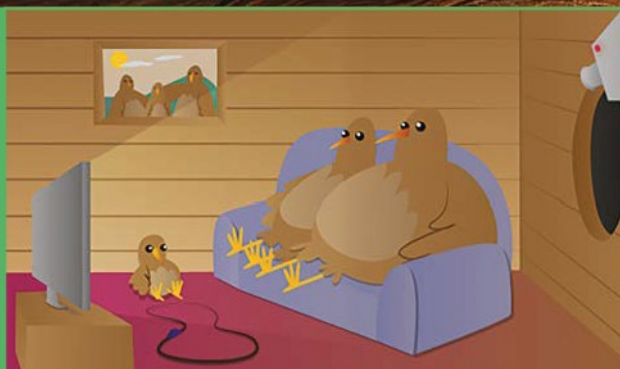
Alloggiato in un contenitore ermetico Tupperware, questo progetto fa uso di una telecamera a infrarossi con LED IR per illuminare la notte. Non solo, ma ti insegna a configurare MotionEyeOS per percepire il movimento e scattare foto e/o video – potrai personalizzare qualcosa se hai in mente di modificare la build.

Come My Naturewatch, fa uso di un contenitore per alimenti a tenuta d'aria, anche se non siamo così sicuri di quanto sia resistente alle intemperie la fotocamera in quella configurazione. Se hai una IR camera ma non una con LED IR, puoi trovare dei LED IR come singoli componenti nella maggior parte dei negozi di elettronica online.

▲ La colla a caldo è in realtà un buon isolante contro il maltempo

◀ È difficile da vedere grazie alla mimetizzazione

BIRDBOX

Un progetto della Raspberry Pi Foundation (**magpi.cc/birdbox**), questa fototrappola usa alcuni concetti che puoi trovare negli altri progetti qui, ma questo comporta anche costruire una casetta per uccelli – un classico progetto di falegnameria per principianti.

Fai attenzione però a assicurarti che qualsiasi progetto come questo non disturberà gli uccelli se nidificano nella scatola.

MONOFILM A600 300 MONOFILM A600

Sembra che un modo popolare per creare la rua foto trappola per animali selvatici sia ospitare l'elettronica in un contenitore ermetico Tupperware e, a quanto pare, usano un grande bottiglia di plastica di soda come una sorta di scudo meteo per l'obiettivo della fotocamera. Almeno, è così che piace costruire le loro macchine fotografiche al team di My Naturewatch.

La parte principale del processo è la costruzione dell'involucro, poiché My Naturewatch ha una immagine per la scheda SD che puoi caricare su Raspberry Pi, permettendoti anche di collegare un telefono o altro dispositivo Wi-Fi, in modo da poterla controllare e anche vedere gli scatti.

Se stai cercando di costruirne uno, Pimoroni vende anche un kit con tutta l'elettronica che serve, qui: **magpi.cc/mynaturekit**.

My Naturewatch
mynaturewatch.net

▲ Puoi navigare nella fotocamera con un telefono o altro dispositivo

◀ La soluzione è molto fai-da-te, ma funziona!

Antarctic PiCam
magpi.cc/130

▲ L'hardware è attaccato a una slitta di legno che scivola nel tubo in PVC

▶ È stato utilizzato in acque antartiche per studiare le alghe ghiacciate

MONOFILM A600 300 MONOFILM A600

Ti va di scattare foto sott'acqua? Il progetto Antarctic PiCam, presente nel numero 130 (**magpi.cc/130**), è stato creato da uno studente e fa uso di una guarnizione idraulica molto semplice, serrato da un dado, per creare un involucro impermeabile. Con una lastra acrilica trasparente a una estremità, la fotocamera può puntare verso l'esterno e riprendere filmati. Dal punto di vista del software, è un po' più grezzo nella sua configurazione originaria, ma utilizzando alcune delle tecniche di My Naturewatch, la semplice trappola fotografica dovrebbe fare qualcosa di più utile per la fotografia naturalistica.

Attenzione!
Elettroutensili

Sii consapevole dei problemi causati da utensili elettrici. Attenzione alle vibrazioni, forti rumori e usa indumenti protettivi e occhiali.
magpi.cc/powertools

kit Completo

Andiamo al sodo con Naturebytes

Naturebytes Wildlife Cam
naturebytes.org
137$ / 125Euro

Scopri tutto sulla astrofotografia in *MagPi 128* **(bit.ly/MagPi128It)**

MONOFILM A600 300 MONOFILM A600

La Wildlife Cam di Naturebytes è un dispositivo basato su Raspberry Pi A+ che viene fornito con una custodia fantastica che contiene Raspberry Pi, un Raspberry Pi Camera Module, una lente IR per ottimizzare il rilevamento del movimento e persino una certificazione IP55 di resistenza alle intemperie. Ciò significa che può proteggere da quantità di polvere che comprometterebbe il funzionamento e anche da un getto d'acqua, anche se si spera che tu non lo colpisca con la tua idropulitrice.

Dovrai assemblarlo da solo. ma richiede solo cacciaviti e le istruzioni sono molto complete con un sacco di foto - e ci sono anche alcune videoguide. Inoltre, puoi cambiare il Raspberry Pi A+ per uno Zero o un Modello B, così come altri tipi di Camera Module se vuoi davvero migliorare la qualità delle foto.

Puoi anche ottenere il solo involucro a metà prezzo se vuoi personalizzarlo ulteriormente. Il kit esiste oramai da molti anni ed è diventato un classico kit Raspberry Pi.

▲ I componenti del kit Naturebtes

Altra fotografia all'aperto

Cos'altro puoi vedere dal tuo giardino?

Camera All Sky economica
magpi.cc/cheapskycam

MONOFILM A600 300 MONOFILM A600

Usando una semplice impermeabilizzazione come un vaso per piante e una cupola, questa semplice build può aiutarti a tenere traccia delle meteore nel cielo usando del codice per rilevare il movimento. Utilizza una lente fish eye a 180° per vedere il cielo e le foto risultanti includono meteore che attraversano il cielo e scie stellari in time-lapse.

▲ Le scie stellari sono uno stile di astrofotografia assolutamente da provare

JULY IN NATURE

Nel Regno Unito, luglio è il primo mese completo d'Estate, quindi vedrai più animaletti e creature in giro. Inclusi i pipistrelli, la lucertola comune, i pulcini di falco pescatore, ermellini e molti tipi di farfalla. Guarda su **discoverwildlife.com** per maggiori info su quel che puoi incontrare nel tuo giardino – e oltre! - durante l'anno.

Credit: Andy Morffew

Timelapse crescita piante
magpi.cc/planttimelapse

MONOFILM A600 300 MONOFILM A600

Questo progetto non solo fornisce acqua alle tue pinate quando necessario, realizza anche una registrazione time-lapse della loro crescita. Questa versione specifica è utilizzata in una serra ma, con qualche esperta modifica (e della impermeabilizzazione), potrebbe essere utilizzata in un giardino o anche per piante da interno. In realtà è una configurazione abbastanza semplice, ma il time-lapse è molto divertente.

▲ La build richiede però una batteria da auto.